

Antonio Carlos Manucci Pereira Júnior

✘ Aluno de Mestrado pelo Programa de Pós-Graduação em Bioquímica do Instituto de Química da USP

Formado em Ciências Biológicas pela Universidade Federal de São Carlos (campus São Carlos), com período de intercâmbio na Università degli Studi di Milano, Itália. Seu interesse atual é no estudo dos sinais químicos e biomecânicos da matriz extracelular e como a célula os interpreta e responde a eles. Seu projeto de mestrado consiste na construção de um mapa de interações (interactoma) da proteína YAP (co-ativadora transcricional de genes relacionados à proliferação e responsiva à rigidez da matriz) em um modelo tridimensional de acinogênese da glândula mamária, realizado no Laboratório de Sinalização da Matrix Extracelular (E-signal Lab, IQ-USP).

Contato: acmpjr@gmail.com | acmpjr@iq.usp.br

Site do Laboratório: [E-Signal Lab](#)

Perfis Acadêmico-Científicos: [Currículo Lattes](#) | [Research Gate](#) | [Biblioteca Virtual Fapesp](#)

Redes Sociais: [Facebook](#) | [Instagram](#)

Meus Exercícios

Link para a página com os exercícios feitos no decorrer do curso:

- [exec](#)
-

Trabalho Final

Propostas de Trabalho

Proposta Principal: Cálculo da eficiência de amplificação de um gene-alvo

A reação em cadeia da polimerase (PCR) é uma ferramenta poderosa em biologia molecular por permitir a amplificação de sequências específicas de DNA ou cDNA milhares ou milhões de vezes de maneira relativamente simples. Diferentemente de uma PCR convencional, na qual o produto é analisado apenas ao fim da reação, na PCR quantitativa (qPCR) a quantidade de produto formado pode ser medida a cada ciclo utilizando sondas ou marcadores fluorescentes, sendo que o seu monitoramento durante a fase de amplificação exponencial da reação nos permite determinar de maneira bastante precisa a quantidade inicial da sequência-alvo da amplificação, o que torna a qPCR a técnica mais utilizada para calcular de maneira relativa os níveis de expressão de um determinado

gene.

Entre inúmeros parâmetros definidos em uma reação de *qPCR*, existe um, denominado eficiência de amplificação, que nos informa o quão eficiente os *primers* utilizados são em amplificar a sequência de interesse e é fundamental tanto para definir a quantidade de cDNA total a ser utilizada na reação quanto para corrigir dados na quantificação relativa nos níveis de expressão gênica.

Sendo assim, a proposta é criar uma função que calcule a eficiência de amplificação do gene-alvo para um ou mais pares de *primers* utilizando os valores de *Ct* obtidos do registros do termociclador (*cycle threshold*, valor de um ponto no qual a curva da reação de amplificação intersecta o *threshold line*, e pode ser definido como o número de ciclos que demora para que se detecte um sinal real das amostras). Em um experimento de eficiência utiliza-se uma série de concentrações de cDNA (que normalmente varia com triplicatas de 10pg a 100ng por volume de reação, e uma triplicata de concentração 0 para controle). Quanto maior a quantidade de cDNA utilizada, menor será o valor de *Ct*, e a partir disso é possível estabelecer uma correlação entre estas duas variáveis.

De maneira geral, a função irá tratar como variável independente uma série de concentrações de cDNA e como variável-resposta os valores de *Ct* registrados para gerar um modelo linear entre elas; o *slope* do modelo será utilizado para calcular a eficiência de amplificação do par de *primers*. Essa função será particularmente útil para realizar o cálculo para inúmeros *primers* de uma só vez como em estudos do efeito de genes reguladores, em que é necessário quantificar a expressão de todo o conjunto de genes influenciados pela expressão do gene regulador (que em alguns casos pode chegar a mais de 30 genes).

Os argumentos de entrada da função serão um *dataframe* com variáveis de classe numérica (no qual obrigatoriamente a primeira coluna deve corresponder à variável independente, e o restante corresponderá à variável-resposta), um valor numérico para o limite de detecção de contaminação observados nos controles da reação e um valor lógico para retirada de valores NA, caso presentes.

A saída da função será um *dataframe* contendo: a) uma coluna com os nomes dos primers utilizados; b) uma coluna com os valores do *intercept* obtidos para cada modelo; c) uma coluna com os valores do *slope* obtidos para cada modelo; d) uma coluna contendo os valores da “eficiência de amplificação” calculados de acordo com a fórmula $E = 10^{(-1/s)}$, onde “E” é o parâmetro a ser estipulado e “s” é o *slope* do modelo; e) uma coluna contendo o resultado da verificação de qualidade dos controles da reação.

Referências:

- Life Technologies. Real-time PCR handbook. Disponível em: [Life Technologies](#)
- Agilent Technologies. Introduction to Quantitative PCR. Disponível em: [Agilent Technologies](#)
- Pfaffl, M. W. A new mathematical model for relative quantification in real-time RT-PCR. *Nucleic Acids Research*, 2001, vol. 29, n. 9:00.

Olá Antonio,

Sua função é interessante, porém é um pouco confusa. Você pretende calcular a eficiência de amplificação de um ou mais primers em uma *qPCR*. Esclarece o que você quer dizer com

quais seriam as réplicas das variáveis de resposta, para as quais imagino você teria uma observação pareada da variável preditora. Você pretende ajustar modelos de regressão simples para cada par da variável preditora vs variável resposta, ou um modelo único de regressão múltipla? Isso não ficou claro para mim. Esclarece um pouco melhor os elementos de entrada e saída da função por favor. Por ex., o argumento numérico que especifica a concentração do primer será um único número? É um vetor? Vai de quanto a quanto? Ou seja, que valores pode tomar? Outra coisa, na saída imagino que vc colocará os objetos numa lista (com exceção do gráfico que irá para o dispositivo gráfico), certo? A eficiência de amplificação é um valor único? Tenta na correção da proposta fazer um esquema de quais seriam especificamente os argumentos da função e seus valores, e os objetos de saída. — [Gustavo Agudelo](#)
2017/06/03 19:46

Proposta Alternativa: Cálculo da concentração proteica

Determinar de maneira precisa a concentração de proteínas presentes em uma amostra é um procedimento comum na rotina de laboratórios de Bioquímica e indispensável para inúmeros protocolos experimentais. Existem muitos métodos desenvolvidos para a dosagem do conteúdo proteico de uma solução e cada um deles possui a sua sensibilidade, compatibilidade química e aplicação. Apesar dessas diferenças, no entanto, a maioria se baseia em uma reação colorimétrica entre um reagente químico e uma amostra da solução a ser testada seguida pela leitura da absorbância em um determinado comprimento de onda (o qual é método-específico); a partir do valor da absorbância obtido pode-se calcular a concentração proteica seguindo a lei de Lambert-Beer, a qual estabelece uma relação linear (dentro de um *range*) entre a absorbância de uma solução e a sua concentração, quando atravessada por uma radiação luminosa.

Sendo assim, a proposta é criar uma função que calcule a concentração proteica de uma ou mais amostras utilizando os valores de absorbância obtidos pela leitura em um espectrofotômetro, a partir da comparação com uma curva padrão de absorbância para amostras de concentrações conhecidas. De maneira geral, a função irá tratar dados de uma variável independente e de réplicas de uma variável-resposta e gerar um modelo de regressão linear, sendo que o *slope* e o *intercept* do modelo, que definem a relação entre as duas variáveis, serão utilizados para calcular a relação entre dados de outra variável-resposta com a mesma variável independente.

Os argumentos de entrada da função serão dois *dataframes* com variáveis de classe numérica (sendo o primeiro definindo os valores de absorbância para cada amostra de concentração conhecida e o segundo contendo os valores de absorbância das amostras a serem as concentrações calculadas).

A saída da função será composta por: a) um *scatterplot* da variável-resposta em função do log da variável independente, contendo a reta do modelo de regressão e o intervalo de confiança da reta; b) um objeto contendo o sumário do modelo linear; c) um objeto contendo os parâmetros *intercept* e *slope* do modelo linear; d) um objeto contendo um *dataframe* com as variáveis “amostra” (categórica)

e “concentração” (numérica), calculada a partir da fórmula $C = (abs-s)/i$, onde “C” é a concentração a ser estipulada, “abs” é o valor de absorbância medido, “s” é o *slope* do modelo e “i” é o *intercept* do modelo.

Antonio,

Essa sua outra função não tem muito de diferencial (em termos de programação) com respeito a sua proposta anterior. Praticamente você propõe fazer quase a mesma coisa em diferentes contextos teóricos. Mais do que focar-se no contexto teórico em que a função seria importante, esse contexto deve ser posto ao serviço de uma função mais genérica e de interesse para mais usuários. De novo, em termos de programação não vejo um diferencial do que poderia ser feito por um ajuste rápido de modelos lineares e sua representação gráfica. Uma coisa que você omite nas duas propostas é o diagnóstico dos modelos, e os valores do coeficiente de determinação (R^2). Sugiro você continuar com a proposta A depois de seguir as sugestões. — [Gustavo Agudelo](#) 2017/06/03 19:46

Página de Ajuda da Função

Arquivo da Documentação:

ef.primr

package: nenhum

R Documentation

Cálculo de Eficiência de Amplificação

Description:

Função que calcula a eficiência de amplificação de um gene-alvo para um ou mais pares de primers, utilizando uma série de concentrações de cDNA e os valores de Ct obtidos do registros do termociclador.

Usage:

```
ef.primr(x, ct.ctrl=0.75, na.rm=TRUE)
```

Arguments:

x

um objeto do R de classe "data.frame", sendo que x[1] deve conter a série de concentrações de cDNA utilizada no experimento.

ct.ctrl

valor numérico para o nível de detecção de amplificação inespecífica observada nos controles (sem adição de cDNA).

na.rm

um valor lógico indicando se valores NA presentes devem ser retirados antes dos cálculos. Os dados gerados do experimentos não possuem valores NA, a menos que o usuário os tenha inserido durante a manipulação prévia da planilha.

Details:

A função realiza o cálculo da eficiência baseando-se no slope da regressão linear dos valores de Ct em função do logaritmo da série de concentrações de cDNA utilizadas no experimento. O cálculo da eficiência é feito de acordo com a fórmula $E=10^{(-1/s)}$, onde E é a eficiência de amplificação de um gene por um dado par de primer e s é o slope do modelo de regressão. Além disso, a função também testa a amplificação observada no controle baseado no argumento inserido na função, sugerindo a possibilidade ou não de contaminação dos primers com DNA exógeno.

Value:

A função retorna um objeto da classe data.frame contendo os nomes dos primers, os valores de intercept e slope do modelo criado para cada par de primer (podem ser utilizados para inserir a linha de tendência em um gráfico utilizando a função abline()), o valor da eficiência de amplificação calculado para cada par de primer e uma coluna contendo a verificação de contaminação do primer, sendo "ok" para uma média de amplificação do controle da reação abaixo do limite aceitável e "check.for.contamination" para a possibilidade de contaminação dos primers.

Warning:

A função retorna uma mensagem de erro "Objeto deve ser de classe data.frame" e para a função quando o objeto inserido na função não pertence à classe "data.frame".

Author(s):

Antonio Carlos Manucci Pereira Júnior

NUSP 9902617

References:

- Life Technologies. Real-time PCR handbook. Disponível em:
- Agilent Technologies. Introduction to Quantitative PCR. Disponível em:
- Pfaffl, M. W. A new mathematical model for relative quantification in real-time RT-PCR. Nucleic Acids Research, 2001, vol. 29, n. 9:00.

Examples:

##exemplo 1:

```
cdna.amount <- rep( c(0,10,100,1000,10000,100000), each=3)      #valores
aparecem em notação (0e+00, 1e+01, 2e+02) mas a função funciona.
gapdh <-
c(0.15,0.3,0.25,32.7,33.4,32.9,29.5,28.7,28.9,25.5,25.3,24.9,21.4,20.9,21.1,
16.5,17.5,17.1)
b.actin <-
c(0.1,0.2,0.1,35.2,35.2,34.9,32.2,32.0,32.4,27.6,27.9,27.3,22.7,23.2,22.9,19
.1,18.6,18.7)
yap1 <-
c(3.7,3.4,3.6,30.7,30.4,30.9,27.5,26.7,27.9,23.5,23.3,23.9,19.4,19.9,19.1,15
.8,15.5,16.1)
teste <- data.frame(cdna.amount,gapdh,b.actin,yap1)      #cria o data.frame
para ser usado na função.

ef.primr(teste, ct.ctrl=1)
```

##exemplo 2:

```
gapdh <-
c(0.15,0.3,0.25,32.7,33.4,32.9,29.5,28.7,28.9,25.5,25.3,24.9,21.4,20.9,21.1,
16.5,17.5,17.1)

ef.primr(gapdh)
```

Código da Função

```
ef.primr <- function(x, ct.ctrl=0.75, na.rm=TRUE)
{
  if(class(x)!="data.frame")      #verificação se o objeto inserido à
função pertence à classe "data.frame".
  {
    stop("\t", "Objeto deve ser de classe data.frame")      #informar uma
mensagem de erro quando o objeto não é um "data.frame".
  }
  if(na.rm==TRUE)      #verificação se valores NA devem ser retirados do
cálculo.
```

```
{
  na.omit(x)      #retira eventuais valores NA presentes.
}
data.x <- x[x[,1] > 0,]      #cria um dataframe retirando apenas os
valores de cDNA maiores que 0.
slp <- rep(NA, length=ncol(x))      #cria um vetor para armazenar os
valores do slope obtidos na função for() a seguir.
intcpt <- rep(NA, length=ncol(x))      #cria um vetor para armazenar os
valores do intercept obtidos na função for() a seguir.
for(i in 2:(ncol(x)))      #função for() para a geração dos modelos para
cada par de primer e obtenção dos valores de intercept e slope.
{
  lin.mod <- lm(data.x[,i] ~ log(data.x[,1], base=10))      #cria o
modelo linear dos valores de Ct em função da coluna 1 do
data.frame(concentrações de cDNA).
  coef.vector <- coef(lin.mod)      #atribui a um objeto os
coeficientes da regressão da rodada(intercept e slope).
  slp[i] <- coef.vector[2]      #armazena no vetor criado anteriormente
os valores de slope de cada rodada da função for().
  intcpt[i] <- coef.vector[1]      #armazena no vetor criado
anteriormente os valores de slope de cada rodada da função for().
}
slope <- slp[-1]      #retira o valor de posição 1 (ocupado por um NA) do
vetor slp.
intercept <- intcpt[-1]      #retira o valor de posição 1 (ocupado por um
NA) do vetor intcpt.
amplification.efficiency <- 10^(-1/slope)      #realiza o cálculo da
eficiência de amplificação.

datactrl.x <- x[x[,1] == 0,]      #cria um dataframe contendo apenas os
valores de cDNA iguais a 0, os quais teoricamente não devem apresentar
amplificação.
ctrl <- rep(NA, length=ncol(x))      #cria um vetor para armazenar a
média dos Ct de cada primer para cDNA=0.
for(j in 2:(ncol(x)))      #função for() para a média dos valores de Ct
de cada primer para cDNA=0.
{
  ctrl[j] <- mean(datactrl.x[,j])      #cálculo da média de Ct de cada
primer para cDNA=0.
}
ctrl.d <- ctrl[-1]      #retira o valor da posição 1 (ocupado por um NA)
do vetor ctrl.
quality.control <- ifelse(ctrl.d <= ct.ctrl, "ok",
"check.for.contamination")      #testa se a média é maior/menor que o
argumento ct.ctrl inserido
primer.name <- colnames(data.x[-1])      #cria um vetor com só com os
nomes dos primers.
output <- data.frame(primer.name, intercept, slope,
amplification.efficiency, quality.control)      #cria um data.frame com os
vetores especificados
return(output)      #retorna o data.frame criado, como resultado da
```

```
função .  
}
```

Arquivo da Função

Formato .R: [função ef.primr](#)

Formato .txt: [função ef.primr](#)

Planilha de Exemplo (.txt): [Teste-qPCR](#)

From:
<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2017:alunos:trabalho_final:acmpjr:start

Last update: **2020/08/12 06:04**

